The Natural Language Decathlon: Multitask Learning as Question Answering

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, Richard Socher
Salesforce Research
{bmccann,nkeskar,cxiong,rsocher}@salesforce.com

Abstract

Deep learning has improved performance on many natural language processing (NLP) tasks individually. However, general NLP models cannot emerge within a paradigm that focuses on the particularities of a single metric, dataset, and task. We introduce the Natural Language Decathlon (decaNLP), a challenge that spans ten tasks: question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, relation extraction, goal-oriented dialogue, semantic parsing, and commonsense pronoun resolution. We cast all tasks as question answering over a context. Furthermore, we present a new multitask question answering network (MQAN) that jointly learns all tasks in decaNLP without any task-specific modules or parameters. MQAN shows improvements in transfer learning for machine translation and named entity recognition, domain adaptation for sentiment analysis and natural language inference, and zero-shot capabilities for text classification. We demonstrate that the MQAN's multi-pointer-generator decoder is key to this success and that performance further improves with an anti-curriculum training strategy. Though designed for decaNLP, MQAN also achieves state of the art results on the WikiSQL semantic parsing task in the single-task setting. We also release code for procuring and processing data, training and evaluating models, and reproducing all experiments for decaNLP.

1 Introduction

We introduce the Natural Language Decathlon (decaNLP) in order to explore models that generalize to many different kinds of NLP tasks. decaNLP encourages a single model to simultaneously optimize for ten tasks: question answering, machine translation, document summarization, semantic parsing, sentiment analysis, natural language inference, semantic role labeling, relation extraction, goal oriented dialogue, and pronoun resolution.

We frame all tasks as question answering [Kumar et al., 2016] by allowing task specification to take the form of a natural language question q: all inputs have a context, question, and answer (Fig. 1). Traditionally, NLP examples have inputs x and outputs y, and the underlying task t is provided through explicit modeling constraints. Meta-learning approaches include t as additional input [Schmidhuber, 1987, Thrun and Pratt, 1998, Thrun, 1998, Vilalta and Drissi, 2002]. Our approach does not use a single representation for any t, but instead uses natural language questions that provide descriptions for underlying tasks. This allows single models to effectively multitask and makes them more suitable as pretrained models for transfer learning and meta-learning: natural language questions allow a model to generalize to completely new tasks through different but related task descriptions.

We provide a set of baselines for decaNLP that combine the basics of sequence-to-sequence learning [Sutskever et al., 2014, Bahdanau et al., 2014, Luong et al., 2015b] with pointer networks [Vinyals et al., 2015, Merity et al., 2017, Gülçehre et al., 2016, Gu et al., 2016, Nallapati et al., 2016], ad-

Preprint. Work in progress.

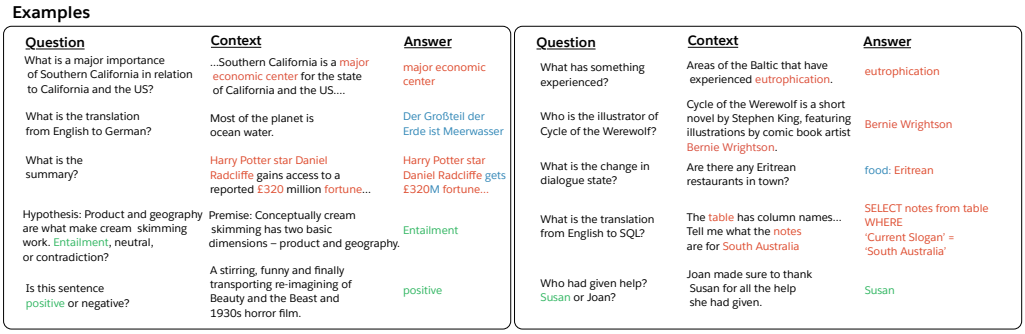

Figure 1: Overview of the decaNLP dataset with one example from each decaNLP task in the order presented in Section 2. They show how the datasets were pre-processed to become question answering problems. Answer words in red are generated by pointing to the context, in green from the question, and in blue if they are generated from a classifier over the output vocabulary.

vanced attention mechanisms [Xiong et al., 2017], attention networks [Vaswani et al., 2017], question answering [Seo et al., 2017, Xiong et al., 2018, Yu et al., 2016, Weissenborn et al., 2017], and curriculum learning [Bengio et al., 2009].

The multitask question answering network (MQAN) is designed for decaNLP and makes use of a novel dual coattention and multi-pointer-generator decoder to multitask across all tasks in decaNLP. Our results demonstrate that training the MQAN jointly on all tasks with the right anti-curriculum strategy can achieve performance comparable to that of ten separate MQANs, each trained separately. A MQAN pretrained on decaNLP shows improvements in transfer learning for machine translation and named entity recognition, domain adaptation for sentiment analysis and natural language inference, and zero-shot capabilities for text classification. Though not explicitly designed for any one task, MQAN proves to be a strong model in the single-task setting as well, achieving state-of-the-art results on the semantic parsing component of decaNLP.

We have released code[1] for obtaining and preprocessing datasets, training and evaluating models, and tracking progress through a leaderboard based on decathlon scores (decaScore). We hope that the combination of these resources will facilitate research in multitask learning, transfer learning, general embeddings and encoders, architecture search, zero-shot learning, general purpose question answering, meta-learning, and other related areas of NLP.

2 Tasks and Metrics

decaNLP consists of 10 publicly available datasets with examples cast as (question, context, answer) triplets as shown in Fig. 1.

Question Answering. Question answering (QA) models receive a question and a context that contains information necessary to output the desired answer. We use the Stanford Question Answering Dataset (SQuAD) [Rajpurkar et al., 2016] for this task. Contexts are paragraphs taken from the English Wikipedia, and answers are sequences of words copied from the context. SQuAD uses a normalized F1 (nF1) metric that strip out articles and punctuation.

Machine Translation. Machine translation models receive an input document in a source language that must be translated into a target language. We use the 2016 English to German training data prepared for the International Workshop on Spoken Language Translation (IWSLT) [Cettolo et al., 2016]. Examples are from transcribed TED presentations that cover a wide variety of topics with conversational language. We evaluate with a corpus-level BLEU score [Papineni et al., 2002] on the 2013 and 2014 test sets as validation and test sets, respectively.

Summarization. Summarization models take in a document and output a summary of that document. Most important to recent progress in summarization was the transformation of the CNN/DailyMail (CNN/DM) corpus [Hermann et al., 2015] into a summarization dataset [Nallapati et al., 2016]. We

[1] https://github.com/salesforce/decaNLP

Table 1: Summary of openly available benchmark datasets in decaNLP and evaluation metrics that contribute to the decaScore. All metrics are case insensitive. nF1 is the normalized F1 metric used by SQuAD that strips out articles and punctuation. EM is an exact match comparison: for text classification, this amounts to accuracy; for WOZ it is equivalent to turn-based dialogue state exact match (dsEM) and for WikiSQL it is equivalent to exact match of logical forms (lfEM). F1 for QA-ZRE is a corpus level metric (cF1) that takes into account that some question are unanswerable. Precision is the true positive count divided by the number of times the system returned a non-null answer. Recall is the true positive count divided by the number of instances that have an answer.

Task	Dataset	# Train	# Dev	# Test	Metric
Question Answering	SQuAD	87599	10570	9616	nF1
Machine Translation	IWSLT	196884	993	1305	BLEU
Summarization	CNN/DM	287227	13368	11490	ROUGE
Natural Language Inference	MNLI	392702	20000	20000	EM
Sentiment Analysis	SST	6920	872	1821	EM
Semantic Role Labeling	QA-SRL	6414	2183	2201	nF1
Zero-Shot Relation Extraction	QA-ZRE	840000	600	12000	cF1
Goal-Oriented Dialogue	WOZ	2536	830	1646	dsEM
Semantic Parsing	WikiSQL	56355	8421	15878	lfEM
Pronoun Resolution	MWSC	80	82	100	EM

include the non-anonymized version of this dataset in decaNLP. On average, these examples contain the longest documents in decaNLP and force models to balance extracting from the context with generation of novel, abstractive sequences of words. CNN/DM uses ROUGE-1, ROUGE-2, and ROUGE-L scores [Lin, 2004]. We average these three measures to compute an overall ROUGE score.

Natural Language Inference. Natural Language Inference (NLI) models receive two input sentences: a premise and a hypothesis. Models must then classify the inference relationship between the two as one of entailment, neutrality, or contradiction. We use the Multi-Genre Natural Language Inference Corpus (MNLI) [Williams et al., 2017] which provides training examples from multiple domains (transcribed speech, popular fiction, government reports) and test pairs from seen and unseen domains. MNLI uses an exact match (EM) score.

Sentiment Analysis. Sentiment analysis models are trained to classify the sentiment expressed by input text. The Stanford Sentiment Treebank (SST) [Socher et al., 2013] consists of movie reviews with the corresponding sentiment (positive, neutral, negative). We use the unparsed, binary version [Radford et al., 2017]. SST also uses an EM score.

Semantic Role Labeling. Semantic role labeling (SRL) models are given a sentence and predicate (typically a verb) and must determine 'who did what to whom,' 'when,' and 'where' [Johansson and Nugues, 2008]. We use an SRL dataset that treats the task as question answering, QA-SRL [He et al., 2015]. This dataset covers both news and Wikipedia domains, but we only use the latter in order to ensure that all data for decaNLP can be freely downloaded. We evaluate QA-SRL with the nF1 metric used for SQuAD.

Relation Extraction. Relation extraction systems take in a piece of unstructured text and the kind of relation that is to be extracted from that text. In this setting, it is important that models can report that the relation is not present and cannot be extracted. As with SRL, we use a dataset that maps relations to a set of questions so that relation extraction can be treated as question answering: QA-ZRE [Levy et al., 2017]. Evaluation of the dataset is designed to measure zero shot performance on new kinds of relations – the dataset is split so that relations seen at test time are unseen at train time. This kind of zero-shot relation extraction, framed as question answering, makes it possible to generalize to new relations. QA-ZRE uses a corpus-level F1 metric (cF1) in order to accurately account for unanswerable questions. This F1 metric defines precision as the true positive count divided by the number of times the system returned a non-null answer and recall as the true positive count divided by the number of instances that have an answer.

Goal-Oriented Dialogue. Dialogue state tracking is a key component of goal-oriented dialogue systems. Based on user utterances, actions taken already, and conversation history, dialogue state trackers keep track of which predefined goals the user has for the dialogue system and which kinds of requests the user makes as the system and user interact turn-by-turn. We use the English Wizard of

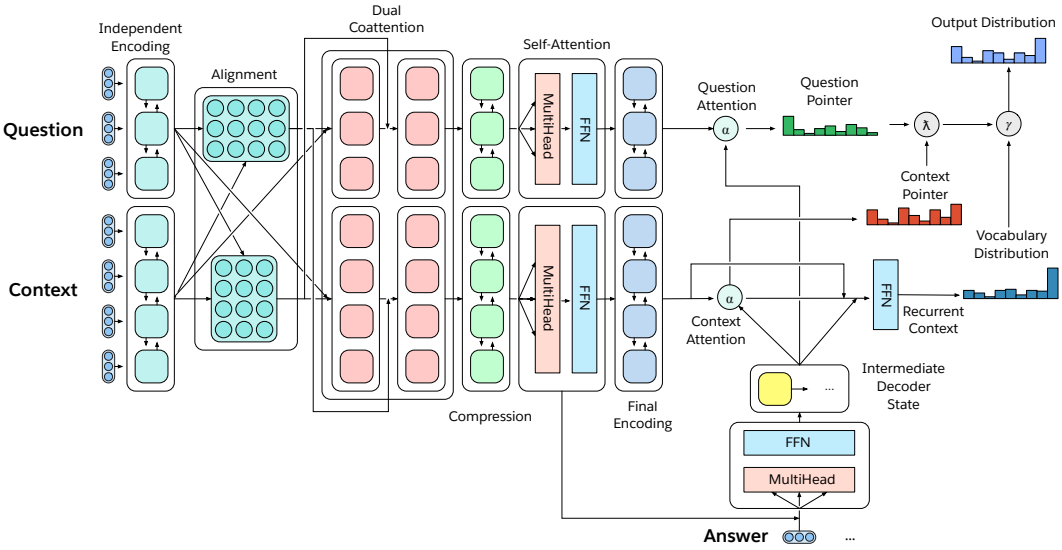

Figure 2: Overview of the MQAN model. It takes in a question and context document, encodes both with a BiLSTM, uses dual coattention to condition representations for both sequences on the other, compresses all of this information with another two BiLSTMs, applies self-attention to collect long-distance dependency, and then uses a final two BiLSTMs to get representations of the question and context. The multi-pointer-generator decoder uses attention over the question, context, and previously output tokens to decide whether to copy from the question, copy from the context, or generate from a limited vocabulary.

Oz (WOZ) restaurant reservation task [Wen et al., 2016], which comes with a predefined ontology of foods, dates, times, addresses, and other information that would help an agent make a reservation for a customer. WOZ is evaluated by turn-based dialogue state EM (dsEM) over the goals of the customers.

Semantic Parsing. SQL query generation is related to semantic parsing. Models based on the WikiSQL dataset [Zhong et al., 2017] translate natural language questions into structured SQL queries so that users can interact with a database in natural language. WikiSQL is evaluated by a logical form exact match (lfEM) to ensure that models do not obtain correct answers from incorrectly generated queries.

Pronoun Resolution. Our final task is based on Winograd schemas [Winograd, 1972], which require pronoun resolution: "Joan made sure to thank Susan for the help she had [given/received]. Who had [given/received] help? Susan or Joan?". We started with examples taken from the Winograd Schema Challenge [Levesque et al., 2011] and modified them to ensure that answers were a single word from the context. This modified Winograd Schema Challenge (MWSC) ensures that scores are neither inflated nor deflated by oddities in phrasing or inconsistencies between context, question, and answer. We evaluate with an EM score.

The Decathlon Score (decaScore). Models competing on decaNLP are evaluated using an additive combination of each task-specific metric. All metrics fall between 0 and 100, so that the decaScore naturally falls between 0 and 1000 for ten tasks. Using an additive combination avoids issues that arise from weighing different metrics. All metrics are case insensitive.

3 Multitask Question Answering Network (MQAN)

Because every task is framed as question answering and trained jointly, we call our model a multitask question answering network (MQAN). Each example consists of a context, question, and answer as shown in Fig. 1. Many recent QA models for question answering typically assume the answer can be copied from the context [Wang and Jiang, 2017, Seo et al., 2017, Xiong et al., 2018], but this assumption does not hold for general question answering. The question often contains key information that constrains the answer space. Noting this, we extend the coattention of [Xiong et al., 2017] to

enrich the representation of not only the input but also the question. Also, the pointer-mechanism of [See et al., 2017] is generalized into a hierarchical, multi-pointer-generator that enables the capacity to copy directly from the question and the context.

During training, the MQAN takes as input three sequences: a context c with l tokens, a question q with m tokens, and an answer a with n tokens. Each of these is represented by a matrix where the ith row of the matrix corresponds to a d_{emb}-dimensional embedding (such as word or character vectors) for the ith token in the sequence:

$$C \in \mathbb{R}^{l \times d_{emb}} \qquad Q \in \mathbb{R}^{m \times d_{emb}} \qquad A \in \mathbb{R}^{n \times d_{emb}} \tag{1}$$

An encoder takes these matrices as input and uses a deep stack of recurrent, coattentive, and self-attentive layers to produce final representations, $C_{fin} \in \mathbb{R}^{l \times d}$ and $Q_{fin} \in \mathbb{R}^{m \times d}$, of both context and question sequences designed to capture local and global interdependencies. Appendix C describes the full details of the encoder.

Answer Representations. During training, the decoder begins by projecting the answer embeddings onto a d-dimensional space:

$$AW_2 = A_{proj} \in \mathbb{R}^{n \times d} \tag{2}$$

This is followed by a self-attentive layers, which has a corresponding self-attentive layer in the encoder. Because it lacks both recurrence and convolution, we add to A_{proj} positional encodings [Vaswani et al., 2017] $PE \in \mathbb{R}^{n \times d}$ with entries

$$PE[t,k] = \begin{cases} \sin(t/10000^{k/2d}) & k \text{ is even} \\ \cos(t/10000^{(k-1)/2d}) & k \text{ is odd} \end{cases} \qquad A_{proj} + PE = A_{ppr} \in \mathbb{R}^{n \times d} \tag{3}$$

Multi-head Decoder Attention. We use self-attention[2] [Vaswani et al., 2017] so that the decoder is aware of previous outputs (or a special intialization token in the case of no previous outputs) and attention over the context to prepare for the next output. Refer to Appendix C for definitions of MultiHead attention and FFN, the residual feedforward network applied after MultiHead attention over the context.

$$\text{MultiHead}_A(A_{ppr}, A_{ppr}, A_{ppr}) = A_{mha} \in \mathbb{R}^{n \times d} \tag{4}$$

$$\text{MultiHead}_A C((A_{mha} + A_{ppr}), C_{fin}, C_{fin}) = A_{ac} \in \mathbb{R}^{n \times d} \tag{5}$$

$$FFN_A(A_{ac} + A_{mha} + A_{ppr}) = A_{self} \in \mathbb{R}^{n \times d} \tag{6}$$

Intermediate Decoder State. We next use a standard LSTM with attention to get a recurrent context state $\tilde{c}_t$ for time-step t. First, the LSTM produces an intermediate state h_t using the previous answer word A_{self}^{t-1} and recurrent context state [Luong et al., 2015b]:

$$\text{LSTM}([(A_{self})_{t-1}; \tilde{c}_{t-1}], h_{t-1}) = h_t \in \mathbb{R}^d \tag{7}$$

Context and Question Attention. This intermediate state is used to get attention weights α_t^C and α_t^Q to allow the decoder to focus on encoded information relevant to time step t.

$$\text{softmax}\left(C_{fin}(W_2 h_t)\right) = \alpha_t^C \in \mathbb{R}^l \qquad \text{softmax}\left(Q_{fin}(W_3 h_t)\right) = \alpha_t^Q \in \mathbb{R}^m \tag{8}$$

Recurrent Context State. Context representations are combined with these weights and fed through a feedforward network with tanh activation to form the recurrent context state and question state:

$$\tanh\left(W_4 \left[C_{fin}^\top \alpha_t^C; h_t\right]\right) = \tilde{c}_t \in \mathbb{R}^d \qquad \tanh\left(W_5 \left[Q_{fin}^\top \alpha_t^Q; h_t\right]\right) = \tilde{q}_t \in \mathbb{R}^d \tag{9}$$

Multi-Pointer-Generator. Our model must be able to generate tokens that are not in the context or the question. We give it access to v additional vocabulary tokens. We obtain distributions over tokens in the context, question, and this external vocabulary, respectively, as

$$\sum_{i:c_i = w_t} \left(\alpha_t^C\right)_i = p_c(w_t) \in \mathbb{R}^n \qquad \sum_{i:q_i = w_t} \left(\alpha_t^Q\right)_i = p_q(w_t) \in \mathbb{R}^m \tag{10}$$

[2]The decoder operates step by step. To prevent the decoder from seeing future time-steps during training, appropriate entries of $XY^\top$ are set to a large negative number prior to the softmax in (22).

Table 2: Validation metrics for decaNLP baselines: sequence-to-sequence (S2S) with self-attentive transformer layers (w/SAtt), the addition of coattention (+CAtt) over a split context and question, and a question pointer (+QPtr). The last model is equivalent to MQAN. Multitask models use a round-robin batch-level sampling strategy to jointly train on the full decaNLP. The last column includes an additional anti-curriculum (+ACurr) phase that trains on SQuAD alone before switching to the fully joint strategy. Entries marked with '-' would correspond to decaScores for aggregates of separately trained models; this is not well-defined without a mechanism for choosing between models.

	Single-task Training				Multitask Training				
Dataset	S2S	w/SAtt	+CAtt	+QPtr	S2S	w/SAtt	+CAtt	+QPtr	+ACurr
SQuAD	48.2	68.2	74.6	75.5	47.5	66.8	71.8	70.8	74.3
IWSLT	25.0	23.3	26.0	25.5	14.2	13.6	9.0	16.1	13.7
CNN/DM	19.0	20.0	25.1	24.0	25.7	14.0	15.7	23.9	24.6
MNLI	67.5	68.5	34.7	72.8	60.9	69.0	70.4	70.5	69.2
SST	86.4	86.8	86.2	88.1	85.9	84.7	86.5	86.2	86.4
QA-SRL	63.5	67.8	74.8	75.2	68.7	75.1	76.1	75.8	77.6
QA-ZRE	20.0	19.9	16.6	15.6	28.5	31.7	28.5	28.0	34.7
WOZ	85.3	86.0	86.5	84.4	84.0	82.8	75.1	80.6	84.1
WikiSQL	60.0	72.4	72.3	72.6	45.8	64.8	62.9	62.0	58.7
MWSC	43.9	46.3	40.4	52.4	52.4	43.9	37.8	48.8	48.4
decaScore	-	-	-	-	513.6	546.4	533.8	562.7	**571.7**

$$\text{softmax}\left(W_v \tilde{c}_t\right) = p_v(w_t) \in \mathbb{R}^v \tag{11}$$

These distributions are extended to cover the union of the tokens in the context, question, and external vocabulary by setting missing entries in each to 0 so that each distribution is in $\mathbb{R}^{l+m+v}$. Two scalar switches regulate the importance of each distribution in determining the final output distribution.

$$\sigma\left(W_{pv}\left[\tilde{c}_t; h_t; \left(A_{self}\right)_{t-1}\right]\right) = \gamma \in [0,1] \qquad \sigma\left(W_{cq}\left[\tilde{q}_t; h_t; \left(A_{self}\right)_{t-1}\right]\right) = \lambda \in [0,1] \tag{12}$$

$$\gamma p_v(w_t) + (1-\gamma)\left[\lambda p_c(w_t) + (1-\lambda) p_q(w_t)\right] = p(w_t) \in \mathbb{R}^{l+m+v} \tag{13}$$

We train using a token-level negative log-likelihood loss over all time-steps: $\mathcal{L} = -\sum_t^T \log p(a_t)$.

4 Experiments and Analysis

4.1 Baselines and MQAN

In our framework, training examples are (question, context, answer) triplets. Our first baseline is a pointer-generator sequence-to-sequence (S2S) model [See et al., 2017]. S2S models take in only a single input sequence, so we concatenate the context and question for this model. In Table 2, validation metrics reveal that the S2S model does not perform well on SQuAD. On WikiSQL, it obtains a much higher score than prior sequence-to-sequence baselines [Zhong et al., 2017], but it is low compared to MQAN (+QPtr) and the other baselines.

Augmenting the S2S model with self-attentive (w/ SAtt) encoder and decoder layers Vaswani et al. [2017], as detailed in C, increases the model's capacity to integrate information from both context and question. This improves performance on SQuAD by 20 nF1, QA-SRL by 4 nF1, and WikiSQL by 12 LFEM. For WikiSQL, this model nearly matches the prior state-of-the-art validation results of 72.4% without using a structured approach [Dong and Lapata, 2018, Huang et al., 2018, Yu et al., 2018b].

We next explore splitting the context and question into two input sequences and augmenting the S2S model with a coattention mechanism (+CAtt). Performance on SQuAD and QA-SRL increases by more than 5 nF1 each. Unfortunately, this fails to improve other tasks, and it significantly hurts performance on MNLI and MWSC. For these two tasks, answers can be copied directly from the

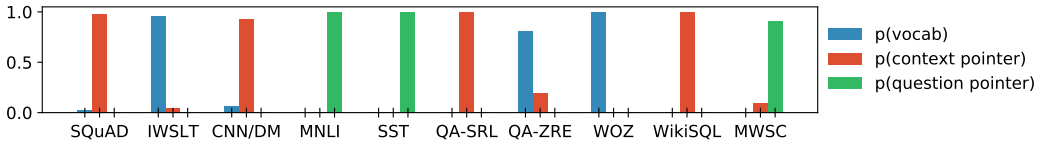

Figure 3: An analysis of how the MQAN chooses to output answer words. When p(generation) is highest, the MQAN places the most weight on the external vocab. When p(context) is highest, the MQAN places the most weight on the pointer distribution over the context. When p(question) is highest, the MQAN places the most weight on the pointer distribution over the question.

question. Because both S2S baselines had the question concatenated to the context, the pointer-generator mechanism was able to copy directly from the question. When the context and question were separated into two different inputs, the model lost this ability.

To remedy this, we add a question pointer (+QPtr) to the previous baseline, which gives the MQAN described in Section 3 and Appendix C. This boosts performance on both MNLI and MWSC above prior baselines. It also improved performance on SQuAD to 75.5 nF1, which matches performance of the first wave of SQuAD models to make use of direct span supervision [Xiong et al., 2017]. This makes it the highest performing question answering model trained on SQuAD dataset that does not explicitly model the problem as span extraction.

This last model achieved a new state-of-the-art test result on WikiSQL by reaching 72.4% lfEM and 80.4% database execution accuracy, surpassing the previous state of the art set by [Dong and Lapata, 2018] at 71.7% and 78.5%.

In the multitask setting, we see similar results, but we also notice several additional striking features. QA-ZRE performance increases 11 F1 points over the highest single-task models, which supports the hypothesis that multitask learning can lead to better generalization for zero-shot learning.

Performance on tasks that require heavy use of the external vocabulary drops more than 50 percent from the S2S baselines until the question pointer is added to the model. In addition to a coattended context, this question pointer makes use of a coattended question, which allows information from the question to flow directly into the decoder. We hypothesize that more direct access to the question makes it easier for the model decide when generating output tokens is more appropriate than copying.

See Appendix B for details regarding pre-processing and hyperparameters.

4.2 Optimization Strategies and Curriculum Learning

For multitask training, we experiment with various round-robin batch-level sampling strategies. Fully joint training cycles through all tasks from the beginning of training. However, some tasks require more iterations to converge in the single-task setting, which suggests that these are more difficult for the model to learn. We experiment with both curriculum and anti-curriculum strategies Bengio et al. [2009] based on this notion of difficulty.

We divide tasks into two groups: the easiest difficult task requires more than twice the iterations the most difficult easy task requires. Compared to the fully joint strategy, curriculum learning jointly trains the easier tasks (SST, QA-SRL, QA-ZRE, WOZ, WikiSQL, and MWSC) first. This leads to a dramatically reduced decaScore (Appendix D). Anti-curriculum strategies boost performance on tasks trained early, but can also hurt performance on tasks held out until later training. Of the various anti-curriculum strategies we experimented with, only the one which trains on SQuAD alone before transitioning to a fully joint strategy yielded a decaScore higher than using the fully joint strategy without modification. For a full comparison, see Appendix D.

4.3 Analysis

Multi-Pointer-Generator and task identification. At each step, the MQAN decides between three choices: generating from the vocabulary, pointing to the question, and pointing to the context. While the model is not trained with *explicit* supervision for these decisions, it learns to switch between the three options. Fig. 3 presents statistics of how often the final model chooses each option. For

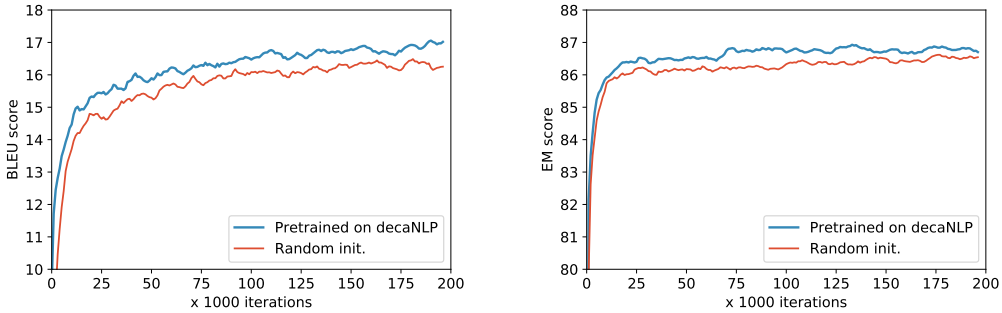

Figure 4: MQAN pretrained on decaNLP outperforms random initialization when adapting to new domains and learning new tasks. Left: training on a new language pair – English to Czech, right: training on a new task – Named Entity Recognition (NER).

SQuAD, QA-SRL, and WikiSQL, the model mostly copies from the context. This is intuitive because all tokens necessary to correctly answer questions from these datasets are contained in the context. The model also usually copies from the context for CNN/DM because answer summaries consist mostly of words from the context with few words generated from outside the context in between.

For SST, MNLI, and MWSC, the model prefers the question pointer because the question contains the tokens for acceptable classes. Because the model learns to use the question pointer in this way, it can do zero-shot classification as discussed in 4.3. For IWSLT and WOZ, the model prefers generating from the vocabulary because German words and dialogue state fields are rarely in the context. The models also avoids copying for QA-ZRE; half of those examples require generating 'unanswerable' from the external vocabulary.

Sampled answers confirm that the model does not confuse tasks. German words are only ever output during translation from English to German. The model never outputs anything but 'positive' and 'negative' for sentiment analysis.

Adaptation to new tasks. MQAN trained ondecaNLP learn to generalize beyond the specific domains for any one task while also learning representations that make learning completely new tasks easier. For two new tasks (English-to-Czech translation and named entity recognition - NER), fine-tuning a MQAN trained on decaNLP requires fewer iterations and reaches a better final performance than training from a random initialization (Fig. 4). For the translation experiment, we use the IWSLT 2016 En→Cs dataset and for NER, we use OntoNotes 5.0 [Hovy et al., 2006].

Zero-shot domain adaptation for text classification. Because MNLI is included in decaNLP, it is possible to adapt to the related Stanford Natural Language Inference Corpus (SNLI) [Bowman et al., 2015]. Fine-tuning a MQAN pretrained on decaNLP achieves an 87% exact match score, which is a 2 point increase over training from a random initialization and 2 points from the state of the art [Kim et al., 2018]. More remarkably, without any fine-tuning on SNLI, a MQAN pretrained on decaNLP still achieves an exact match score of 62%.

Because decaNLP contains SST, it can also perform well on other binary sentiment classification tasks. On Amazon and Yelp reviews [Kotzias et al., 2015], a MQAN pretrained on decaNLP achieves exact match scores of 82.1% and 80.8%, respectively, without any fine-tuning.

Additionally, rephrasing questions by replacing the tokens for the training labels *positive/negative* with *happy/angry* or *supportive/unsupportive* at inference time, leads to only small degradation in performance. The model's reliance on the question pointer for SST (see Figure 3) allows it to copy different, but related class labels with little confusion. This suggests these multitask models are more robust to slight variations in questions and tasks and can generalize to new and unseen classes.

These results demonstrate that models trained on decaNLP have potential to simultanesouly generalize to out-of-domain contexts and questions for multiple tasks and even adapt to unseen classes for text classification. This kind of zero-shot domain adaptation in both input and output spaces suggests that the breadth of tasks in decaNLP encourages generalization beyond what can be achieved by training for a single task.

5 Related Work

This section contains work related to aspects of decaNLP and MQAN that are not task-specific. See Appendix A for work related to each individual task.

Transfer Learning in NLP. Most success in making use of the relatedness between natural language tasks stem from transfer learning. Word2Vec [Mikolov et al., 2013a,b], skip-thought vectors [Kiros et al., 2015] and GloVe [Pennington et al., 2014] yield pretrained embeddings that capture useful information about natural language. The embeddings [Collobert and Weston, 2008, Collobert et al., 2011], intermediate representations [Peters et al., 2018], and weights of language models can be transferred to similar architectures [Ramachandran et al., 2017] and classification tasks [Howard and Ruder, 2018]. Intermediate representations from supervised machine translation models improve performance on question answering, sentiment analysis, and natural language inference [McCann et al., 2017]. Question answering datasets support each other as well as entailment tasks [Min et al., 2017], and high-resource machine translation can support low-resource machine translation [Zoph et al., 2016]. This work shows that the combination of MQAN and decaNLP makes it possible to transfer an entire end-to-end model that can be adapted for any NLP task cast as question answering.

Multitask Learning in NLP. Unified architectures have arisen for chunking, POS tagging, NER, and SRL [Collobert et al., 2011] as well as dependency parsing, semantic relatedness, and natural language inference [Hashimoto et al., 2016]. Multitask learning over different machine translation language pairs can enable zero-shot translation [Johnson et al., 2017], and sequence-to-sequence architectures can be used to multitask across translation, parsing, and image captioning [Luong et al., 2015a] using varying numbers of encoders and decoders. These tasks can also be learned with image classification and speech recognition with careful modularization [Kaiser et al., 2017], and the success of this approach extends to visual and textual question answering [Xiong et al., 2016]. Learning such modularization can further mitigate interference between tasks [Ruder et al., 2017].

More generally, multitask learning has been successful when models are able to capitalize on relatedness amongst tasks while mitigating interference from dissimilarities [Caruana, 1997]. When tasks are sufficiently related, they can provide an inductive bias [Mitchell, 1980] that forces models to learn more generally useful representations. By unifying tasks under a single perspective, it is possible to explore these relationships [Wang et al., 2018, Poliak et al., 2018a,b].

MQAN trained on decaNLP is the first, single model to achieve reasonable performance on such a wide variety of complex NLP tasks without task-specific modules or parameters, with little evidence of catastrophic interference, and without parse trees, chunks, POS tags, or other intermediate representations. This sets the foundation for general question answering models.

Optimization and Catastrophic Forgetting. Multitask learning presents a set of optimization problems that extend beyond the NLP setting. Multi-objective optimization [Deb, 2014] naturally connects to multitask learning and typically involves querying a decision-maker who weighs different objectives. Much effort has gone into mitigating catastrophic forgetting [McCloskey and Cohen, 1989, Ratcliff, 1990, Kemker et al., 2017] by penalizing the norm of parameters when training on a new task [Kirkpatrick et al., 2017], the norm of the difference between parameters for previously learned tasks during parameter updates [Hashimoto et al., 2016], incrementally matching modes [Lee et al., 2017], rehearsing on old tasks [Robins, 1995], using adaptive memory buffers [Gepperth and Karaoguz, 2016], finding task-specific paths through networks [Fernando et al., 2017], and packing new tasks into already trained networks [Mallya and Lazebnik, 2017].

MQAN is able to perform nearly as well or better in the multitask setting as in the single-task setting for each task despite being capped at the same number of trainable parameters in both. A collection of MQANs trained for each task individually would use far more trainable parameters than a single MQAN trained jointly on decaNLP. This suggests that MQAN successfully uses trainable parameters more efficiently in the multitask setting by learning to pack or share parameters in a way that limits catastrophic forgetting.

Meta-Learning Meta-learning attempts to train models on a variety of tasks so that they can easily learn new tasks [Thrun and Pratt, 1998, Thrun, 1998, Vilalta and Drissi, 2002]. Past work has shown how to learn rules for learning [Schmidhuber, 1987, Bengio et al., 1992], train meta-agents that

control parameter updates [Hochreiter et al., 2001, Andrychowicz et al., 2016], augment models with special memory mechanisms [Santoro et al., 2016, Schmidhuber, 1992], and maximize the degree to which models can learn new tasks [Finn et al., 2017].

6 Conclusion

We introduced the Natural Language Decathlon (decaNLP), a new benchmark for measuring the performance of NLP models across ten tasks that appear disparate until unified as question answering. We presented MQAN, a model for general question answering that uses a multi-pointer-generator decoder to capitalize on questions as natural language descriptions of tasks. Despite not having any task-specific modules, we trained MQAN on all decaNLP tasks jointly, and we showed that anti-curriculum learning gave further improvements. After training on decaNLP , MQAN exhibits transfer learning and zero-shot capabilities. When used as pretrained weights, MQAN improved performance on new tasks. It also demonstrated zero-shot domain adaptation capabilities on text classification from new domains. We hope the the decaNLP benchmark, experimental results, and publicly available code encourage further research into general models for NLP.

References

M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, pages 3981–3989, 2016.

J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei. On the optimization of a synaptic learning rule. *Optimality in Artificial and Biological Neural*, pages Networks, pp. , 1992, 1992.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, 2009.

J. Bos and K. Markert. Recognising textual entailment with robust logical inference. In *MLCW*, 2005.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.

R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and M. J. Federico. The iwslt 2016 evaluation campaign. In *IWSLT*, 2016.

M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, et al. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*, 2018.

Q. Chen, X.-D. Zhu, Z.-H. Ling, D. Inkpen, and S. Wei. Natural language inference with external knowledge. *CoRR*, abs/1711.04289, 2017.

J. Choi, K. M. Yoo, and S. goo Lee. Learning to compose task-specific tree structures. In *Association for the Advancement of Artificial Intelligence*, 2017.

R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, 2008.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

C. Condoravdi, D. Crouch, V. de Paiva, R. Stolle, and D. G. Bobrow. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning - Volume 9*, HLT-NAACL-TEXTMEANING '03, 2003.

I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *MLCW*, 2005.

K. Deb. Multi-objective optimization. In *Search methodologies*, pages 403–449. Springer, 2014.

L. Dong and M. Lapata. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*, 2018.

C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Y. Fyodorov, Y. Winter, and N. Francez. A natural logic inference system. In *In Proceedings of the 2nd Workshop on Inference in Computational Semantics (ICoS-2)*, 2000.

J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017.

A. Gepperth and C. Karaoguz. A bio-inspired incremental learning architecture for applied perceptual problems. *Cognitive Computation*, 8(5):924–934, 2016.

R. Ghaeini, S. A. Hasan, V. Datla, J. Liu, K. Lee, A. Qadir, Y. Ling, A. Prakash, X. Z. Fern, and O. Farri. Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. *CoRR*, abs/1802.05577, 2018.

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks : the official journal of the International Neural Network Society*, 18 5-6:602–10, 2005.

M. Grusky, M. Naaman, and Y. Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *NAACL-HLT*, New Orleans, Louisiana, 2018.

J. Gu, Z. Lu, H. Li, and V. O. K. Li. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. *CoRR*, abs/1603.06393, 2016.

c. Gülçehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio. Pointing the Unknown Words. *arXiv preprint arXiv:1603.08148*, 2016.

K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In *ICLR*, 2016.

L. He, M. Lewis, and L. S. Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *EMNLP*, 2015.

L. He, K. Lee, M. Lewis, and L. S. Zettlemoyer. Deep semantic role labeling: What works and what's next. In *ACL*, 2017.

K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9 8:1735–80, 1997.

S. Hochreiter, A. S. Younger, and P. R. Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001.

E. H. Hovy, M. P. Marcus, M. Palmer, L. A. Ramshaw, and R. M. Weischedel. Ontonotes: The 90In *HLT-NAACL*, 2006.

J. Howard and S. Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.

M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou. Reinforced mnemonic reader for machine reading comprehension. *CoRR*, abs/1705.02798, 2018.

H.-Y. Huang, C. Zhu, Y. Shen, and W. Chen. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *CoRR*, abs/1711.07341, 2017.

P.-S. Huang, C. Wang, R. Singh, W.-t. Yih, and X. He. Natural language to structured query generation via meta-learning. *arXiv preprint arXiv:1803.02400*, 2018.

J. Im and S. Cho. Distance-based self-attention network for natural language inference. *CoRR*, abs/1712.02047, 2017.

R. Johansson and P. Nugues. Dependency-based semantic role labeling of propbank. In *EMNLP*, 2008.

M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. S. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351, 2017.

M. Joshi, E. Choi, D. S. Weld, and L. S. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017.

L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. One model to learn them all. *CoRR*, abs/1706.05137, 2017.

R. Kemker, A. Abitino, M. McClure, and C. Kanan. Measuring catastrophic forgetting in neural networks. *CoRR*, abs/1708.02072, 2017.

S. Kim, J.-H. Hong, I. Kang, and N. Kwak. Semantic sentence matching with densely-connected recurrent and co-attentive information. *arXiv preprint arXiv:1805.11360*, 2018.

J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114 13:3521–3526, 2017.

R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015.

D. Kotzias, M. Denil, N. De Freitas, and P. Smyth. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606. ACM, 2015.

A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*, 2016.

S. Lee, J. Kim, J. Jun, J. Ha, and B. Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4655–4665, 2017.

H. J. Levesque, E. Davis, and L. Morgenstern. The winograd schema challenge. In *Aaai spring symposium: Logical formalizations of commonsense reasoning*, volume 46, page 47, 2011.

O. Levy, M. Seo, E. Choi, and L. S. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *CoNLL*, 2017.

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL workshop on Text Summarization*, 2004.

R. Liu, W. Wei, W. Mao, and M. Chikina. Phase conductor on multi-layered attentions for machine comprehension. *CoRR*, abs/1710.10504, 2017a.

X. Liu, Y. Shen, K. Duh, and J. Gao. Stochastic answer networks for machine reading comprehension. *CoRR*, abs/1712.03556, 2017b.

M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114, 2015a.

T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015b.

B. MacCartney and C. D. Manning. An extended model of natural logic. In *IWCS*, 2009.

A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. *arXiv preprint arXiv:1711.05769*, 2017.

D. Marcheggiani, A. Frolov, and I. Titov. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *CoNLL*, 2017.

B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation: Contextualized word vectors. In *NIPS*, 2017.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *ICLR*, 2017.

T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013b.

S. Min, M. J. Seo, and H. Hajishirzi. Question answering through transfer learning from large fine-grained supervision data. In *ACL*, 2017.

T. M. Mitchell. The need for biases in learning generalizations. 1980.

N. Mrkšić, D. O. Séaghdha, T.-H. Wen, B. Thomson, and S. Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

R. Nallapati, B. Zhou, C. N. dos Santos, Çaglar Gülçehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *CoNLL*, 2016.

B. Pan, H. Li, Z. Zhao, B. Cao, D. Cai, and X. He. Memen: Multi-layer embedding with memory networks for machine comprehension. *CoRR*, abs/1707.09098, 2017.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

R. Pasunuru and M. Bansal. Multi-reward reinforced summarization with saliency and entailment. In *NAACL*, 2018.

R. Pasunuru, H. Guo, and M. Bansal. Towards improving abstractive summarization via entailment generation. In *NFiS@EMNLP*, 2017.

R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. S. Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

A. Poliak, Y. Belinkov, J. Glass, and B. Van Durme. On the evaluation of semantic phenomena in neural machine translation using natural language inference. *arXiv preprint arXiv:1804.09779*, 2018a.

A. Poliak, A. Haldar, R. Rudinger, J. E. Hu, E. Pavlick, A. S. White, and B. Van Durme. Towards a unified natural language inference framework to evaluate sentence representations. *arXiv preprint arXiv:1804.08207*, 2018b.

V. Punyakanok, D. Roth, and W. tau Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34:257–287, 2008.

A. Radford, R. Józefowicz, and I. Sutskever. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444, 2017.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.

P. Ramachandran, P. J. Liu, and Q. V. Le. Unsupervised pretraining for sequence to sequence learning. In *EMNLP*, 2017.

R. M. Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97 2:285–308, 1990.

A. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

S. Ruder, J. Bingel, I. Augenstein, and A. Sogaard. Sluice networks: Learning what to share between loosely related tasks. *CoRR*, abs/1705.08142, 2017.

S. Salant and J. Berant. Contextualized word representations for reading comprehension. *CoRR*, abs/1712.03609, 2017.

A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

J. Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL*, 2017.

R. Sennrich, A. Birch, A. Currey, U. Germann, B. Haddow, K. Heafield, A. V. M. Barone, and P. Williams. The university of edinburgh's neural mt systems for wmt17. In *WMT*, 2017.

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *ICLR*, 2017.

T. Shen, T. Zhou, G. Long, J. Jiang, S. Wang, and C. Zhang. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. *CoRR*, abs/1801.10296, 2018.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

J. Suzuki and M. Nagata. Cutting-off redundant repeating generations for neural abstractive summarization. In *EACL*, 2017.

K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi. Deep semantic role labeling with self-attention. *CoRR*, abs/1712.01586, 2017.

Y. Tay, L. A. Tuan, and S. C. Hui. A compare-propagate architecture with alignment factorization for natural language inference. *CoRR*, abs/1801.00102, 2017.

S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

S. Thrun and L. Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. Newsqa: A machine comprehension dataset. In *Rep4NLP@ACL*, 2017.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. In *NIPS*, 2015.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

S. Wang and J. Jiang. Machine comprehension using Match-LSTM and answer pointer. In *ICLR*, 2017.

W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou. Gated self-matching networks for reading comprehension and question answering. In *ACL*, 2017a.

Z. Wang, W. Hamza, and R. Florian. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*, 2017b.

D. Weissenborn, G. Wiese, and L. Seiffe. Making neural qa as simple as possible but not simpler. In *CoNLL*, 2017.

T.-H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426, 2017.

T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. S. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

C. Xiong, S. Merity, and R. Socher. Dynamic Memory Networks for Visual and Textual Question Answering. In *ICML*, 2016.

C. Xiong, V. Zhong, and R. Socher. Dynamic coattention networks for question answering. *ICLR*, 2017.

C. Xiong, V. Zhong, and R. Socher. Dcn+: Mixed objective and deep residual coattention for question answering. *ICLR*, 2018.

A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. Qanet: Combining local convolution with global self-attention for reading compre- hension. In *ICLR*, 2018a.

H. Yu and T. Munkhdalai. Neural semantic encoders. *Proceedings of the conference. Association for Computational Linguistics. Meeting*, 1:397–407, 2017a.

H. Yu and T. Munkhdalai. Neural tree indexers for text understanding. *ACL*, 2017b.

T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. Radev. Typesql: Knowledge-based type-aware neural text-to-sql generation. *arXiv preprint arXiv:1804.09769*, 2018b.

Y. Yu, W. Zhang, K. S. Hasan, M. Yu, B. Xiang, and B. Zhou. End-to-end reading comprehension with dynamic answer chunk ranking. *CoRR*, abs/1610.09996, 2016.

V. Zhong, C. Xiong, and R. Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

V. Zhong, C. Xiong, and R. Socher. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*, 2018.

J. Zhou and W. Xu. End-to-end learning of semantic role labeling using recurrent neural networks. In *ACL*, 2015.

B. Zoph, D. Yuret, J. May, and K. Knight. Transfer learning for low-resource neural machine translation. In *EMNLP*, 2016.

A Further Related Work

Question Answering. Early success on the SQuAD dataset exploited the fact that all answers can be found verbatim in the context. State-of-the-art models point to start and end tokens in the document [Seo et al., 2017, Xiong et al., 2017, Yu et al., 2016, Weissenborn et al., 2017]. This allowed deterministic answer extraction to overtake sequential token generation [Wang and Jiang, 2017]. This quirk of the dataset does not hold for question answering in general, so recent models for SQuAD are not necessarily general question answering models [Yu et al., 2018a, Hu et al., 2018, Wang et al., 2017a, Liu et al., 2017b, Huang et al., 2017, Xiong et al., 2018, Liu et al., 2017a, Pan et al., 2017, Salant and Berant, 2017]. While datasets like TriviaQA [Joshi et al., 2017] and NewsQA [Trischler et al., 2017] could also represent question answering, SQuAD is particularly interesting because the human level performance of SQuAD models in the single-task setting depends on a quirk that does not generalize to all forms of question answering. Including SQuAD in decaNLP challenges models to integrate techniques learned from a single-task approach into a more general approach while evaluation remains grounded in the document. Many of the alternatives are larger and can be used as additional training data or incorporated into future iterations of the decaNLP once the more well-understood SQuAD dataset has been mastered in the multitask setting.

Machine Translation. Until recently, the standard approach trained recurrent models with attention [Luong et al., 2015b, Bahdanau et al., 2014] on a single source-target language pair [Wu et al., 2016, Sennrich et al., 2017]. Models that use only convolution [Gehring et al., 2017] or attention [Vaswani et al., 2017] have shown that recurrence is not essential for the task, but recurrence can contribute to the strongest models [Chen et al., 2018]. While training these models on many source and target languages at the same time remains difficult, limiting models to one source language and many target languages or vice versa can lead to strong performance when resources are limited or null [Johnson et al., 2017].

While much larger corpora and many other language pairs exist, the English-German IWSLT dataset provides the same order of magnitude of training data as the other tasks in decaNLP. We encourage the use of larger corpora or multiple language pairs to improve performance, but we did not want to skew the first iteration of the challenge too far towards machine translation.

Summarization Recent approaches combine recurrent neural networks with pointer networks to generate output sequences that contain key words copied from the document [Nallapati et al., 2016]. Coverage mechanisms [Nallapati et al., 2016, See et al., 2017, Suzuki and Nagata, 2017] and temporal attention [Paulus et al., 2017] improve problems with redundancy in long summaries. Reinforcement learning has pushed performance using common summarization metrics [Paulus et al., 2017] as well as alternative metrics that transfer knowledge from another task [Pasunuru et al., 2017, Pasunuru and Bansal, 2018].

While new corpora like NEWSROOM [Grusky et al., 2018] are even larger, CNN/DM remains the current standard benchmark, so we include it in decaNLP and encourage augmentation with datasets like NEWSROOM.

Natural Language Inference NLI has a long history playing roles in tasks like information retrieval and semantic parsing [Fyodorov et al., 2000, Condoravdi et al., 2003, Bos and Markert, 2005, Dagan et al., 2005, MacCartney and Manning, 2009]. The introduction of the Stanford Natural Language Inference Corpus (SNLI) by [Bowman et al., 2015] spurred a new wave of interest in NLI, its connections to other tasks, and general sentence representations. The most successful approaches make use of attentional models that match and align words in the premise to those in the hypothesis [Tay et al., 2017, Peters et al., 2018, Ghaeini et al., 2018, Chen et al., 2017, Wang et al., 2017b, McCann et al., 2017], but recent non-attentional models designed to extract useful sentence representations have nearly closed the gap [Liu et al., 2017b, Im and Cho, 2017, Shen et al., 2018, Choi et al., 2017].

The dataset we use, the Multi-Genre Natural Language Inference Corpus (MNLI) introduced by [Williams et al., 2017], is the successor to SNLI. Recent approaches to MNLI use methods developed on SNLI and have even pointed out the similarities between models for question answering and NLI [Huang et al., 2017].

Sentiment Analysis Because SST came with parse trees for every example, some approaches use all of the sub-tree labels by modeling trees explicitly [Yu and Munkhdalai, 2017b, Tai et al., 2015] as in the original paper. Others use sub-tree labels implicitly [Yu and Munkhdalai, 2017a, McCann et al., 2017, Peters et al., 2018], and still others do not use the sub-trees at all [Radford et al., 2017]. This suggests that while the many sub-tree labels might facilitate learning, they are not necessary to train state-of-the-art models.

Semantic Role Labeling Traditionally, models have made use of syntactic parsing information Punyakanok et al. [2008], but recent methods have demonstrated that it is not necessary to use syntactic information as additional input [Zhou and Xu, 2015, Marcheggiani et al., 2017]. State-of-the-art approaches treat SRL as a tagging problem [He et al., 2017], make use of that specific structure to constrain decoding, and mix recurrent and self-attentive layers [Tan et al., 2017].

Because QA-SRL treats SRL as question answering [He et al., 2015], it abstracts away the many task-specific constraints of treating SRL as a tagging problem with hand-designed verb-specific roles or grammars. This preserves much of the structure extracted by prior formulations while also allowing models to extract structure that is not syntax-based.

Relation Extraction QA-ZRE introduced a similar idea for relation extraction [Levy et al., 2017]. By associating natural language questions with relations, this dataset reduces relation extraction to question answering. This makes it possible to use question answering models in place of more traditional relation extraction models that often do not make use of the linguistic similarities amongst relations. This in turn makes it possible to do zero-shot relation extraction.

Goal-Oriented Dialogue Dialogue state tracking requires a system to estimate a users goals and and requests given the dialogue context, and it plays a crucial role in goal-oriented dialogue systems. Most models use a structured approach [Mrkšić et al., 2016], with the most recent work making use of both global and local modules to learns representations of the user utterance and previous system actions [Zhong et al., 2018].

Semantic Parsing Similarly, recent approaches to the semantic parsing WikiSQL dataset have made use of structured approaches that move from coarse sketches of the input to fine-grained structured outputs [Dong and Lapata, 2018], direclty employing a type system [Yu et al., 2018b], or making use of dependency graphs [Huang et al., 2018].

B Preprocessing and Training Details

All data is lowercased as is common for SQuAD, IWSLT, CNN/DM, and WikiSQL; casing is irrelevant for the evaluation of the other tasks. We use the RevTok tokenizer[3] to provide simple, yet completely reversible tokenization, which is crucial for detokenizing generated sequences for evaluation. The generative vocabulary in Eq. 11 contains the most frequent 50000 words in the combined training sets for all tasks in decaNLP. SQuAD examples with context longer than 400 tokens were excluded during training and CNN/DM examples had contexts truncated to 400 tokens during training and evaluation. Only MNLI examples with a label other than '-' were included during training and evaluation as is standard. For WOZ, we train turn-by-turn to predict the change in belief state including user requests as an additional slot, but during evaluation we only consider the cumulative belief state as is standard. We do not perform any form of beam search or otherwise refine greedily sampled outputs for any tasks to avoid task-specific post-processing where possible.

The MQAN defined in Section 3 takes 300-dimensional GloVe embeddings trained on Common-Crawl [Pennington et al., 2014] as input. Words that do not have corresponding GloVe embeddings are assigned zero vectors instead. We concatenate 100-dimensional character n-gram embeddings [Hashimoto et al., 2016] to the GloVe embeddings. This corresponds to setting $d_{emb} = 400$ in Section 3. Internal model dimension $d = 200$, hidden dimension $f = 150$, and the number of heads in multi-head attention $p = 3$. MQAN uses 2 self-attention and multi-head decoder attention layers. We use a dropout of 0.2 on inputs to LSTMs, layers following coattention, and decoder layers, before multiplying by $\tilde{Z}$ in Eq. 22, before adding X in Eq. 25, and generally after any linear transformation. The models are trained using Adam with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.98, 10^{-9})$ and a warmup schedule [Vaswani et al., 2017], which increases the learning rate linearly from 0 to 2.5×10^{-3} over 800 iterations before decaying it as $\frac{1}{\sqrt{k}}$, where k is the iteration count. Batches consist entirely of examples from one task and are dynamically constructed to fit as many examples as possible so that the sum of the number of tokens in the context and question and five times the number of tokens in the asnwer does not exceed 10000.

[3] https://github.com/jekbradbury/revtok

C Multitask Question Answering Network (MQAN) Encoder

Recall from Section 3 that the encoder has three input sequences during training: a context c with l tokens, a question q with m tokens, and an answer a with n tokens. Each of these is represented by a matrix where the ith row of the matrix corresponds to a d_{emb}-dimensional embedding (such as word or character vectors) for the ith token in the sequence:

$$C \in \mathbb{R}^{l \times d_{emb}} \qquad Q \in \mathbb{R}^{m \times d_{emb}} \qquad A \in \mathbb{R}^{n \times d_{emb}} \tag{14}$$

Independent Encoding. A linear layer projects input matrices onto a common d-dimensional space.

$$CW_1 = C_{proj} \in \mathbb{R}^{l \times d} \qquad QW_1 = Q_{proj} \in \mathbb{R}^{m \times d} \tag{15}$$

These projected representations are fed into a shared, bidirectional Long Short-Term Memory Network (BiLSTM) [Hochreiter and Schmidhuber, 1997, Graves and Schmidhuber, 2005] [4]

$$\text{BiLSTM}_{ind}(C_{proj}) = C_{ind} \in \mathbb{R}^{l \times d} \qquad \text{BiLSTM}_{ind}(Q_{proj}) = Q_{ind} \in \mathbb{R}^{m \times d} \tag{16}$$

Alignment. We obtain coattended representations by first aligning encoded representations of each sequence. We add separate trained, dummy embeddings to C_{ind} and Q_{ind} (now $\in \mathbb{R}^{(l+1) \times d}$ and $\mathbb{R}^{(m+1) \times d}$) so that tokens are not forced to align with any token in the other sequence.

Let $\text{softmax}\,(X)$ denote a column-wise softmax that normalizes each column of the matrix X to have entries that sum to 1. We obtain alignments by normalizing dot-product similarity scores between representations of one sequence with those of the other:

$$\text{softmax}\left(C_{ind}Q_{ind}^{\top}\right) = S_{cq} \in \mathbb{R}^{(l+1) \times (m+1)} \qquad \text{softmax}\left(Q_{ind}C_{ind}^{\top}\right) = S_{qc} \in \mathbb{R}^{(m+1) \times (l+1)} \tag{17}$$

Dual Coattention. These alignments are used to compute weighted summations of the information from one sequence that is relevant to a single token in the other.

$$S_{cq}^{\top}C_{ind} = C_{sum} \in \mathbb{R}^{(m+1) \times d} \qquad S_{qc}^{\top}Q_{ind} = Q_{sum} \in \mathbb{R}^{(l+1) \times d} \tag{18}$$

The coattended representations use the same weights to transfer information gained from alignments back to the original sequences:

$$S_{qc}^{\top}C_{sum} = C_{coa} \in \mathbb{R}^{(l+1) \times d} \qquad S_{cq}^{\top}Q_{sum} = Q_{coa} \in \mathbb{R}^{(m+1) \times d} \tag{19}$$

The first column of the summation and coattentive representations correspond to the dummy embeddings. This information is not needed, so we drop that column of the matrices to get $C_{coa} \in \mathbb{R}^{l \times d}$ and $Q_{coa} \in \mathbb{R}^{m \times d}$.

Compression. In order to compress information from dual coattention back to the more manageable dimension d, we concatenate all four prior representations for each sequence along the last dimension and feed into separate BiLSTMs:

$$\text{BiLSTM}_{comC}([C_{proj}; C_{ind}; Q_{sum}; C_{coa}]) = C_{com} \in \mathbb{R}^{l \times d} \tag{20}$$

$$\text{BiLSTM}_{comQ}([Q_{proj}; Q_{ind}; C_{sum}; Q_{coa}]) = Q_{com} \in \mathbb{R}^{m \times d} \tag{21}$$

Self-Attention. Next, we use multi-head, scaled dot-product attention [Vaswani et al., 2017] to capture long distance dependencies within each sequence. Let

$$\text{Attention}(\tilde{X}, \tilde{Y}, \tilde{Z}) = \text{softmax}\left(\frac{\tilde{X}\tilde{Y}^{\top}}{\sqrt{d}}\right)\tilde{Z} \tag{22}$$

$$\text{MultiHead}(X, Y, Z) = [h_1; \cdots; h_p]W_o \qquad \text{where } h_j = \text{Attention}(XW_j^X, YW_j^Y, ZW_j^Z) \tag{23}$$

All linear transformations in Eq. (23) project to d so that multi-head attention representations maintain dimensionality:

$$\text{MultiHead}_C(C_{com}, C_{com}, C_{com}) = C_{mha} \qquad \text{MultiHead}_Q(Q_{com}, Q_{com}, Q_{com}) = Q_{mha} \tag{24}$$

[4] For input $X \in \mathbb{R}^{T \times d_{in}}$, let $h_t^{\rightarrow} = \text{LSTM}\left(x_t^T, h_{t-1}^{\rightarrow}\right)$ and $h_t^{\leftarrow} = \text{LSTM}\left(x_t^T, h_{t+1}^{\leftarrow}\right)$. Representations are concatenated along the last dimension $h_t = [h_t^{\rightarrow}; h_t^{\leftarrow}]$ for each t and stacked as rows of output $H \in \mathbb{R}^{T \times d_{out}}$.

We then use projected, residual feedforward networks (FFN) with ReLU activations [Nair and Hinton, 2010, Vaswani et al., 2017] and layer normalization [Ba et al., 2016] on the inputs and outputs. With parameters $U \in \mathbb{R}^{d \times f}$ and $V \in \mathbb{R}^{f \times d}$:

$$FFN(X) = \max(0, XU)V + X \tag{25}$$

$$FFN_C(C_{com} + C_{mha}) = C_{self} \in \mathbb{R}^{l \times d} \qquad FFN_Q(Q_{com} + Q_{mha}) = Q_{self} \in \mathbb{R}^{m \times d} \tag{26}$$

Final Encoding. Finally, we aggregate all of this information across time with two BiLSTMs:

$$\text{BiLSTM}_{finC}(C_{self}) = C_{fin} \in \mathbb{R}^{l \times d} \qquad \text{BiLSTM}_{finQ}(Q_{self}) = Q_{fin} \in \mathbb{R}^{m \times d} \tag{27}$$

These matrices are given to the decoder to generate the answer.

D Curriculum Learning

For multitask training, we experiment with various round-robin batch-level sampling strategies.

The first strategy we consider is fully joint. In this strategy, batches are sampled round-robin from all tasks in a fixed order from the start of training to the end. This strategy performed well on tasks that required fewer iterations to converge during single-task training (see Table 3), but the model struggles to reach single-task performance for several other tasks. In fact, we found a correlation between the performance gap between single and multitasking settings of any given task and number of iterations required for convergence for that task in the single-task setting.

With this in mind, we experimented with several anti-curriculum schedules Bengio et al. [2009]. These training strategies all consist of two phases. In the first phase, only a subset of the tasks are trained jointly, and these are typically the ones that are more difficult. In the second phase, all tasks are trained according to the fully joint strategy.

We first experimented with isolating SQuAD in the first phase, and the switching to fully joint training over all tasks. Since we take a question answering approach to all tasks, we were motivated by the idea of pretraining on SQuAD before being exposed to other kinds of question answering. This would teach the model how to use the multi-context decoder to properly retrieve information from the context before needing to learn how to switch between tasks or generate words on its own. Additionally, pretraining on SQuAD had already been shown to improve performance for NLI [Min et al., 2017]. Empirically, we found that this motivation is well-placed and that this strategy outperforms all others that we considered in terms of the decaScore. This strategy sacrificed performance on IWSLT but recovered the lost decaScore on other tasks, especially those which use pointers.

To explore if adding additional tasks to the initial curriculum would improve performance further, we experimented with adding IWSLT and CNN/DM to the first phase and in another experiment, adding IWSLT, CNN/DM and MNLI. These are tasks with a large number of training examples relative to the other tasks, and they contain the longest answer sequences. Further, they form a diverse set since they encourage the model to decode in different ways such as the vocabulary for IWSLT, context-pointer for SQuAD and CNN/DM, and question-pointer for MNLI. In our results, we however found no improvement by adding these tasks. In fact, in the case when we added SQuAD, IWSLT, CNN/DM and MNLI to the initial curriculum, we observed a marked degradation in performance of some other tasks including QA-SRL, WikiSQL and MWSC. This suggests that it is concordance between the question answering nature of the task and SQuAD that enabled improved outcomes and not necessarily the richness of the task.

Table 3: Validation metrics for MQAN using various training strategies. The first is fully joint, which samples batches round-robin from all tasks. Others first use a curriculum or anti-curriculum schedule over a subset of tasks before switching to fully joint over all tasks. Curriculum first trains tasks that take relatively few iterations to converge when trained alone. This omits SQuAD, IWSLT, CNN/DM, and MNLI. The remaining strategies are anti-curriculum. They include in the first phase either SQuAD alone, SQuAD, IWSLT, and CNN/DM, or SQuAD, IWSLT, CNN/DM, and MNLI.

			Anti-Curriculum		
Dataset	Fully Joint	Curriculum	SQuAD	+IWSLT+CNN/DM	+MNLI
SQuAD	70.8	43.4	74.3	74.5	74.6
IWSLT	16.1	4.3	13.7	18.7	19.0
CNN/DM	23.9	21.3	24.6	20.8	21.6
MNLI	70.5	58.9	69.2	69.6	72.7
SST	86.2	84.5	86.4	83.6	86.8
QA-SRL	75.8	70.6	77.6	77.5	75.1
QA-ZRE	28.0	24.6	34.7	30.1	37.7
WOZ	80.6	81.9	84.1	81.7	85.6
WikiSQL	62.0	68.6	58.7	54.8	42.6
MWSC	48.8	41.5	48.4	34.9	41.5
decaScore	562.7	499.6	571.7	546.2	557.2

Finally, as a check to our hypothesis, we also tried a curriculum schedule that used SST, QA-SRL, QA-ZRE, WOZ, WikiSQL and MWSC in the initial curriculum. This effectively takes the easiest tasks and trains on those first. This was indubitably an inferior strategy; not only does the model perform worse on tasks that were not in the initial curriculum, especially SQuAD and IWSLT, it also performs worse on the tasks that were. Finding that anti-curriculum learning benefited models in the decaNLP also validated intuitions outlined in [Caruana, 1997]: tasks that are easily learned may not lead to development of internal representations that are useful to other tasks. Our results actually suggest a stronger claim: including easy tasks early on in training makes it more difficult to learn internal representations that are useful to other tasks.

We note in passing that the results above underscores the challenges and trade-offs in the multitasking setting. By ordering the tasks differently, it is possible to improve performance on some of the tasks but that improvement is not without a concomitant drop in performance for others. Indeed, a gap still exists between single-task performance and the results above. The question of how this gap can be bridged is a topic of continued research.